Learning Task Specifications for Reinforcement Learning from Human Feedback

David Lindner

Microsoft Swiss JRC Workshop, 29.03.2022

Based on joint work with Matteo Turchetta, Sebastian Tschiatschek, Kamil Ciosek, Katja Hofmann, and Andreas Krause

ETH zürich

Learning & Adaptive Systems

Microsoft

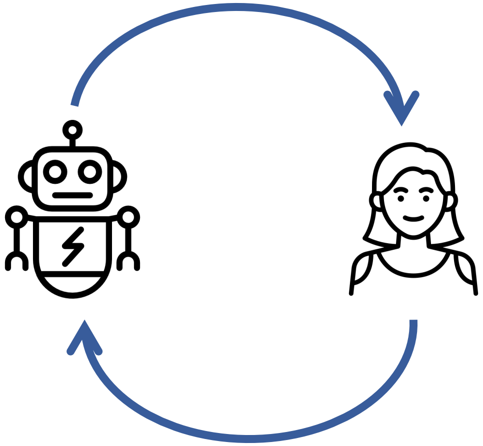
Where do rewards come from?

Well specified
reward function

Autonomous Driving

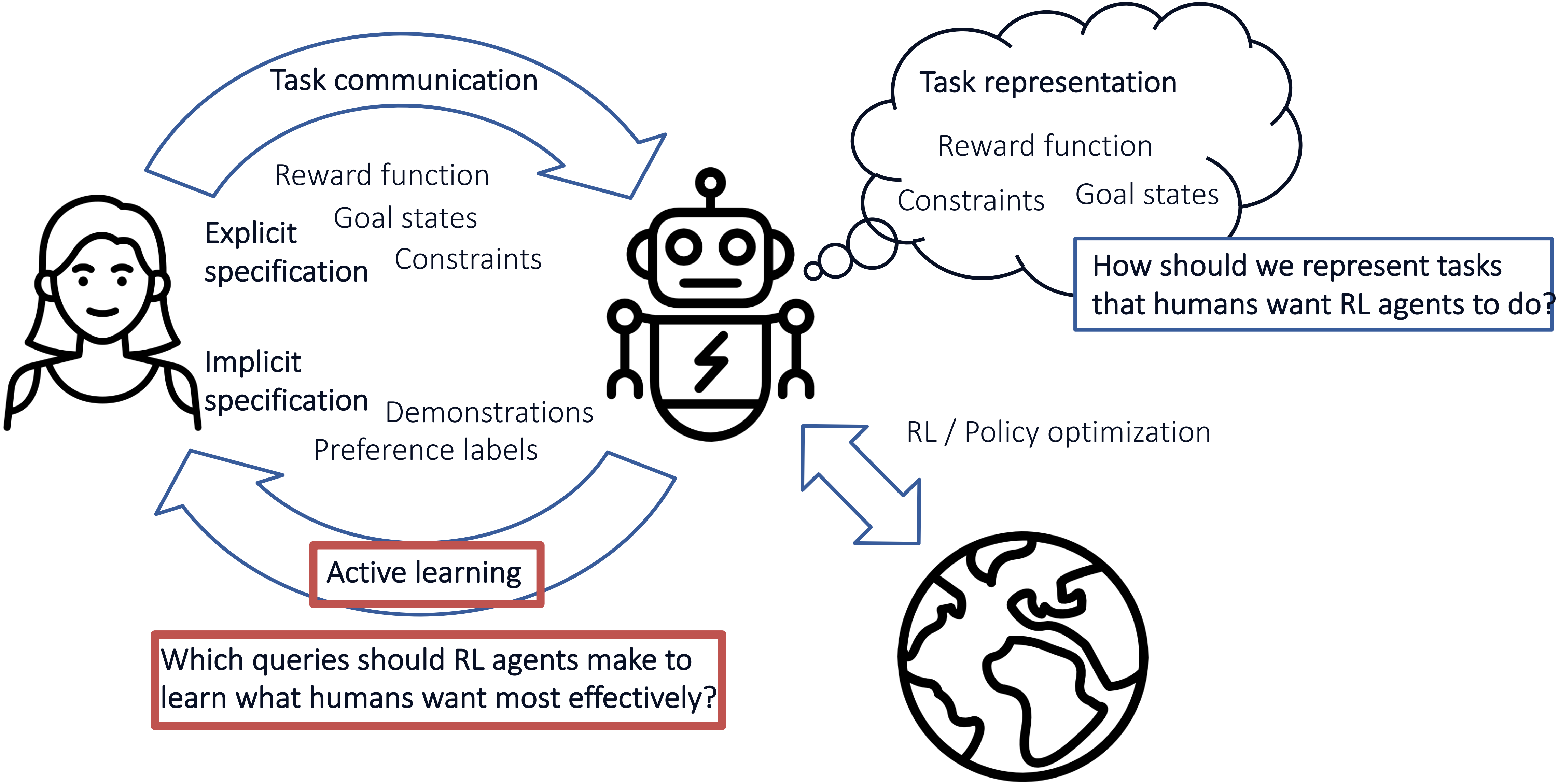
Virtual Assistant

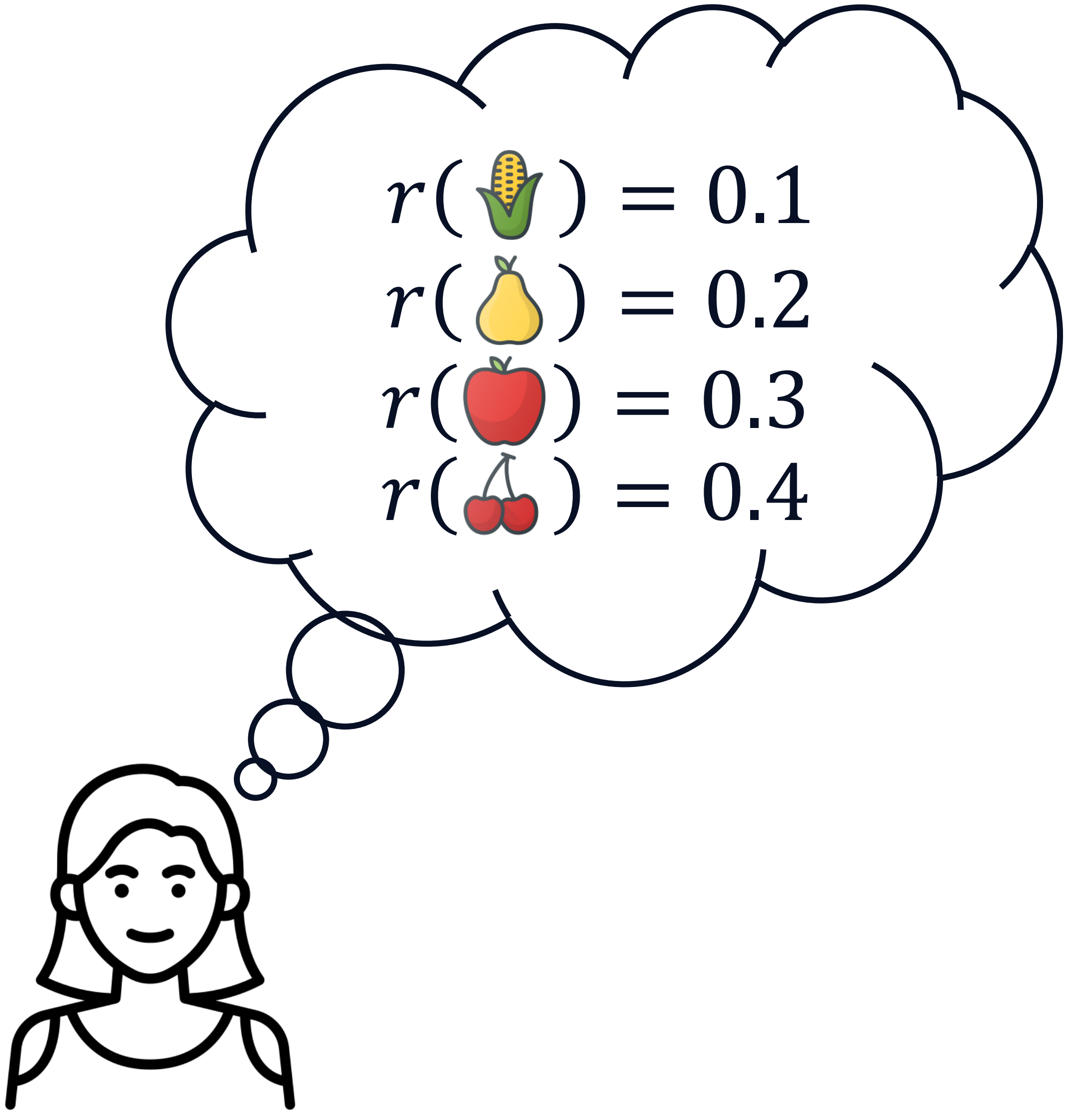
Reward function?

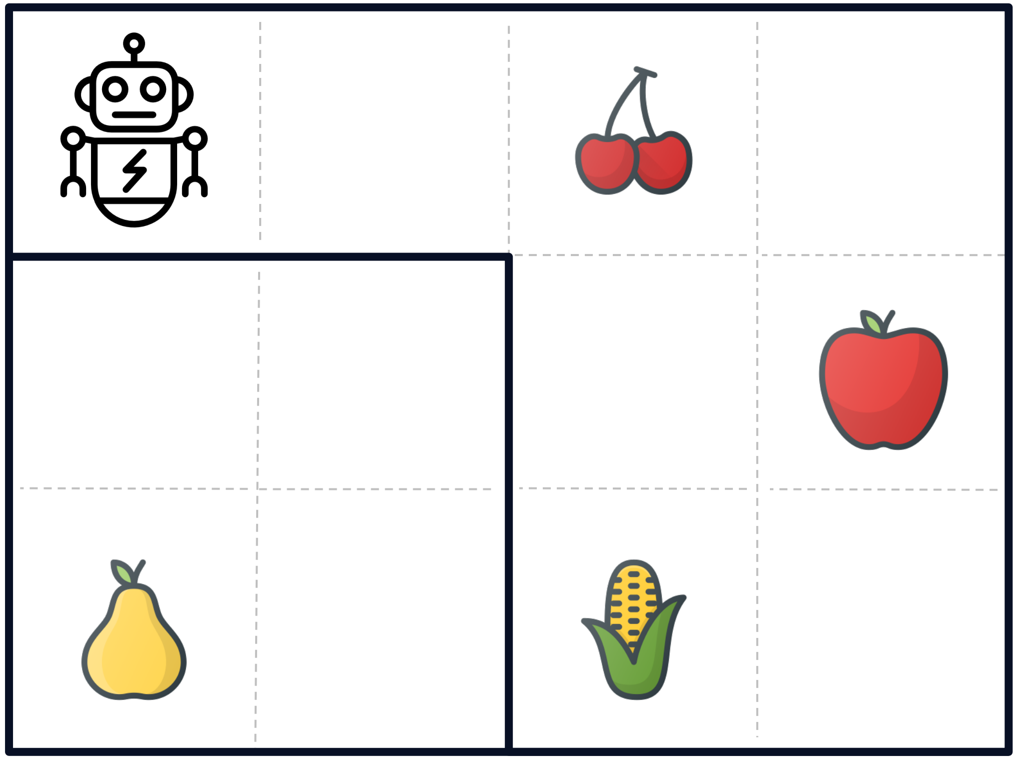


Learning from
human feedback

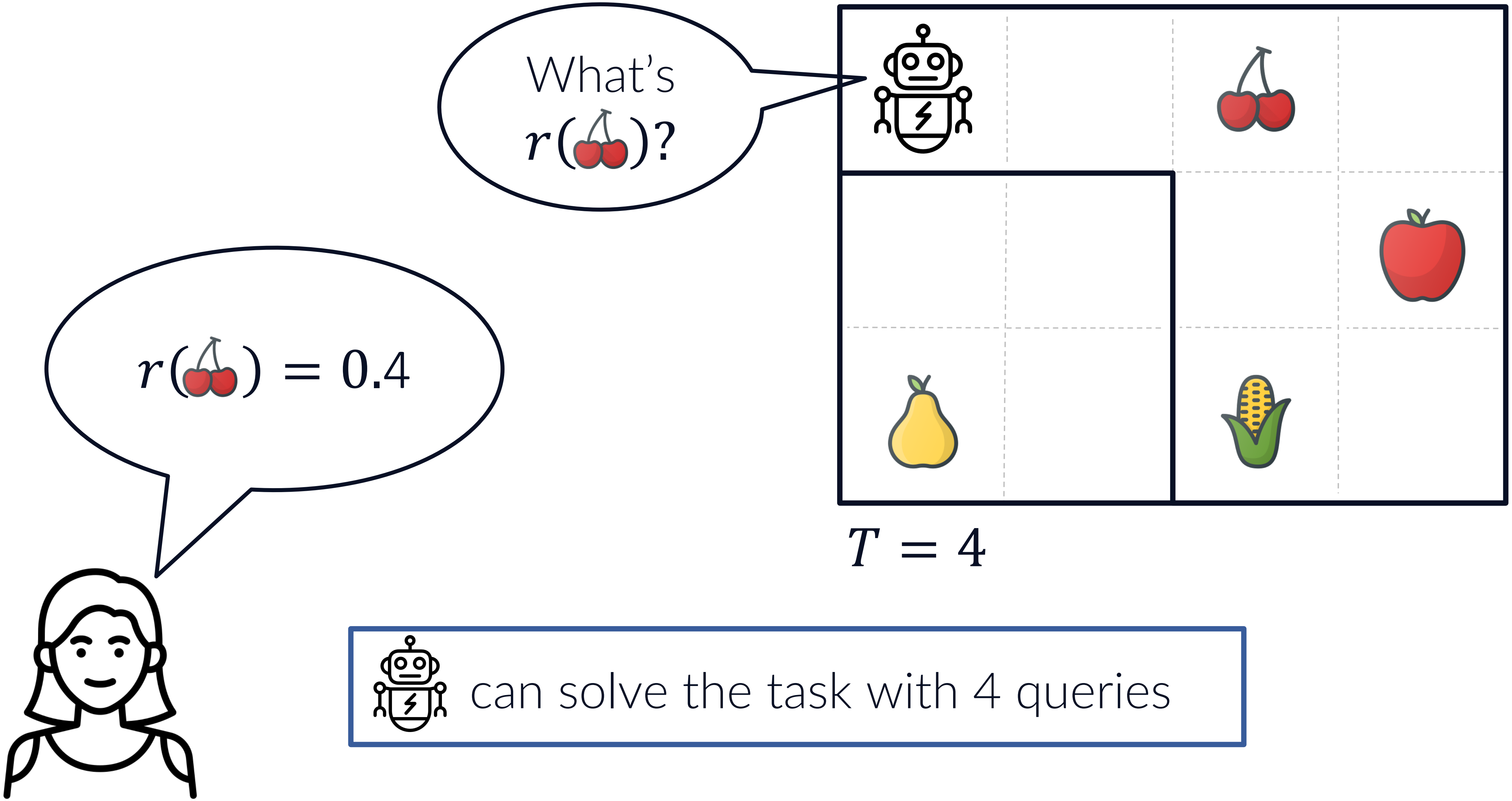
How to communicate task specifications to RL agents?



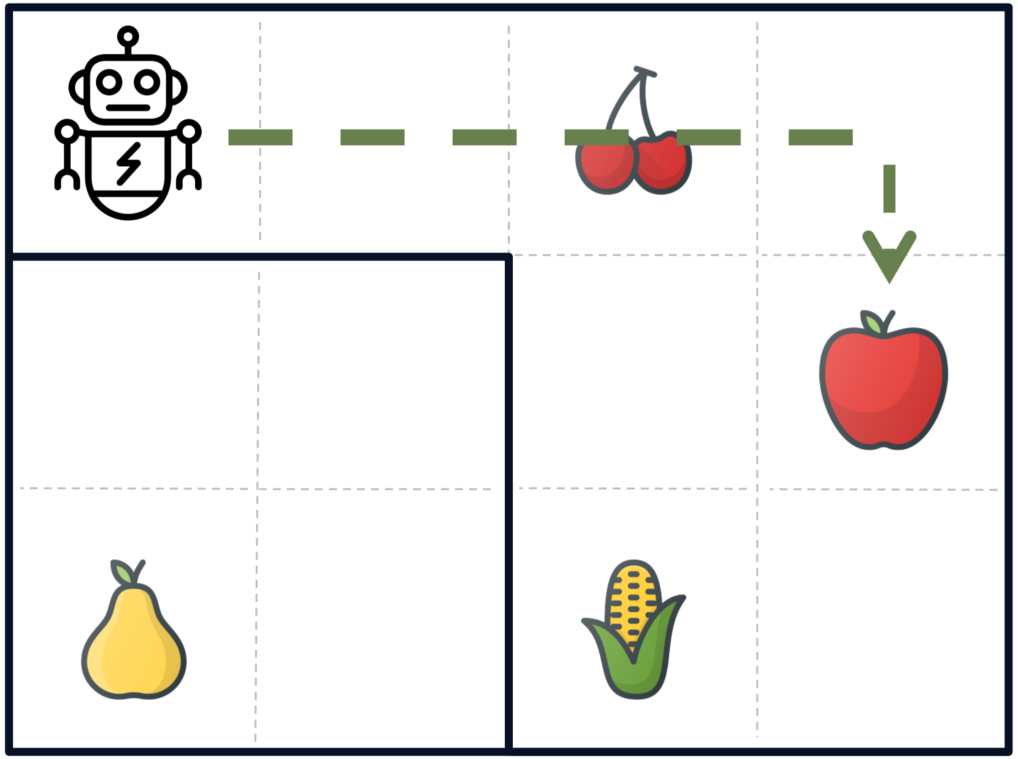


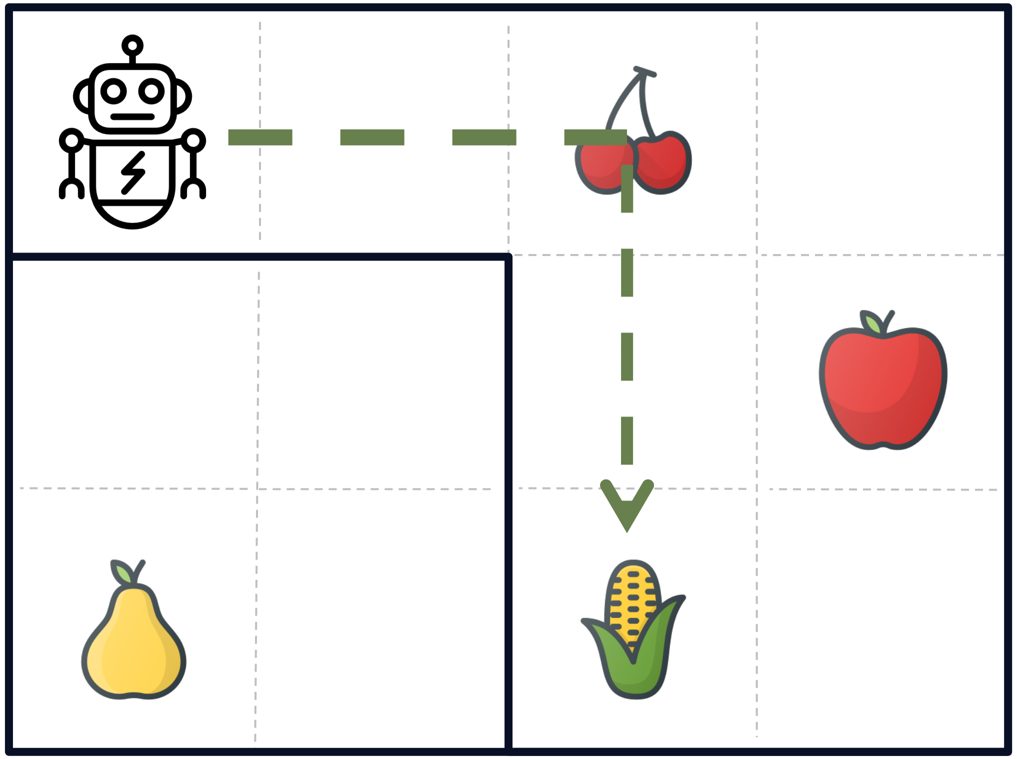


$T = 4$


What’s
r( )?
r( ) = 0.4
T = 4
can solve the task with 4 queries

Considering potentially optimal policies improves sample efficiency





- will always be collected
- will never be collected

only has to decide between and .

How should we select which queries to make?

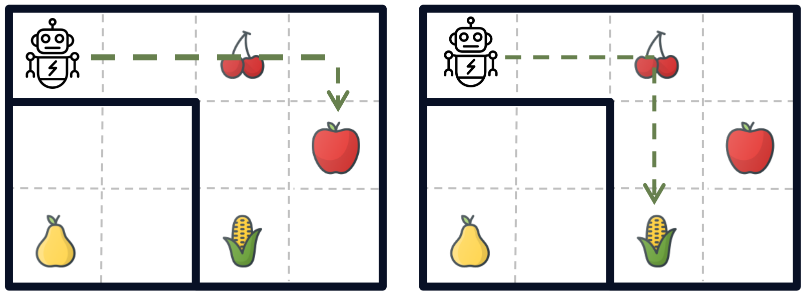
1. Consider potentially optimal policies π_1 and π_2

2. How do they differ?
Difference in return: $\hat{G}(\pi_1) - \hat{G}(\pi_2)$

3. Which observations help most to distinguish them?

$$q^* \in \mathrm{argmax}_{q \in \mathcal{Q}_c} I\left(\hat{G}(\pi_1) - \hat{G}(\pi_2); (q, \hat{y}) \,\middle|\, \mathcal{D}\right)$$

π_1 π_2



$\hat{G}(\pi_1) = \hat{r}(🍒) + \hat{r}(🍎)$

$\hat{G}(\pi_2) = \hat{r}(🍒) + \hat{r}(🌽)$

$\hat{G}(\pi_1) - \hat{G}(\pi_1) = \hat{r}(🍎) - \hat{r}(🌽)$

Relevant states:
🍎 and 🌽

Information directed reward learning (IDRL)

Initialize reward model

Select potentially optimal policies π_1 and π_2

Select query:
$\boldsymbol{q}^* \in \mathbf{argmax}_{q \in \mathcal{Q}_c} \boldsymbol{I}\left(\widehat{\boldsymbol{G}}(\boldsymbol{\pi}_1) - \widehat{\boldsymbol{G}}(\boldsymbol{\pi}_2); (\boldsymbol{q}, \widehat{\boldsymbol{y}}) \middle| \boldsymbol{\mathcal{D}}\right)$

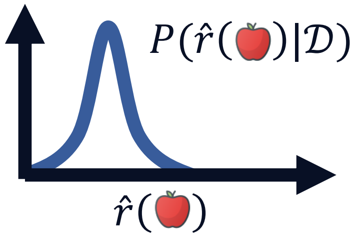
Make query $\boldsymbol{q}^*$
Update reward model

if not done

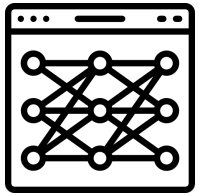
if done

Find and return policy for mean estimate of reward model

Gaussian Process (GP) reward model



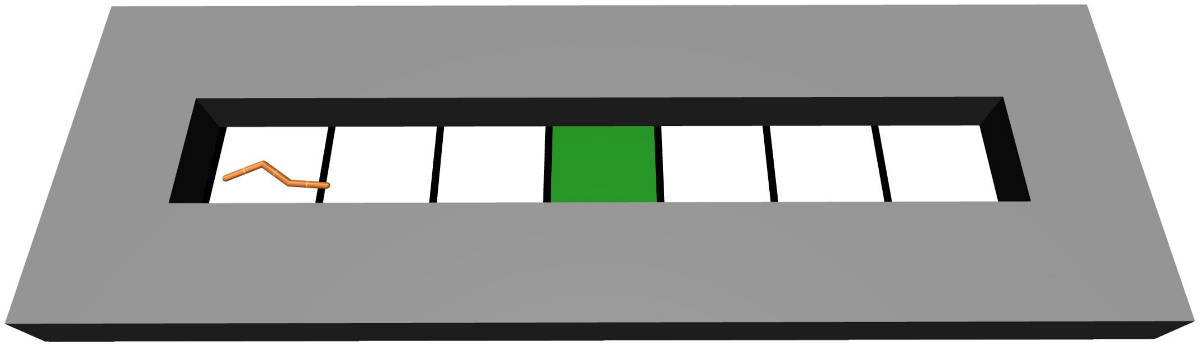
Deep Neural Network reward



- For a GP model we can compute the information gain exactly and efficiently
- For some DNN models, we **can approximate the IDRL objective efficiently**

IDRL can learn a complex task in the MuJoCo simulator from numerical evaluations of clips of trajectories

GP reward model



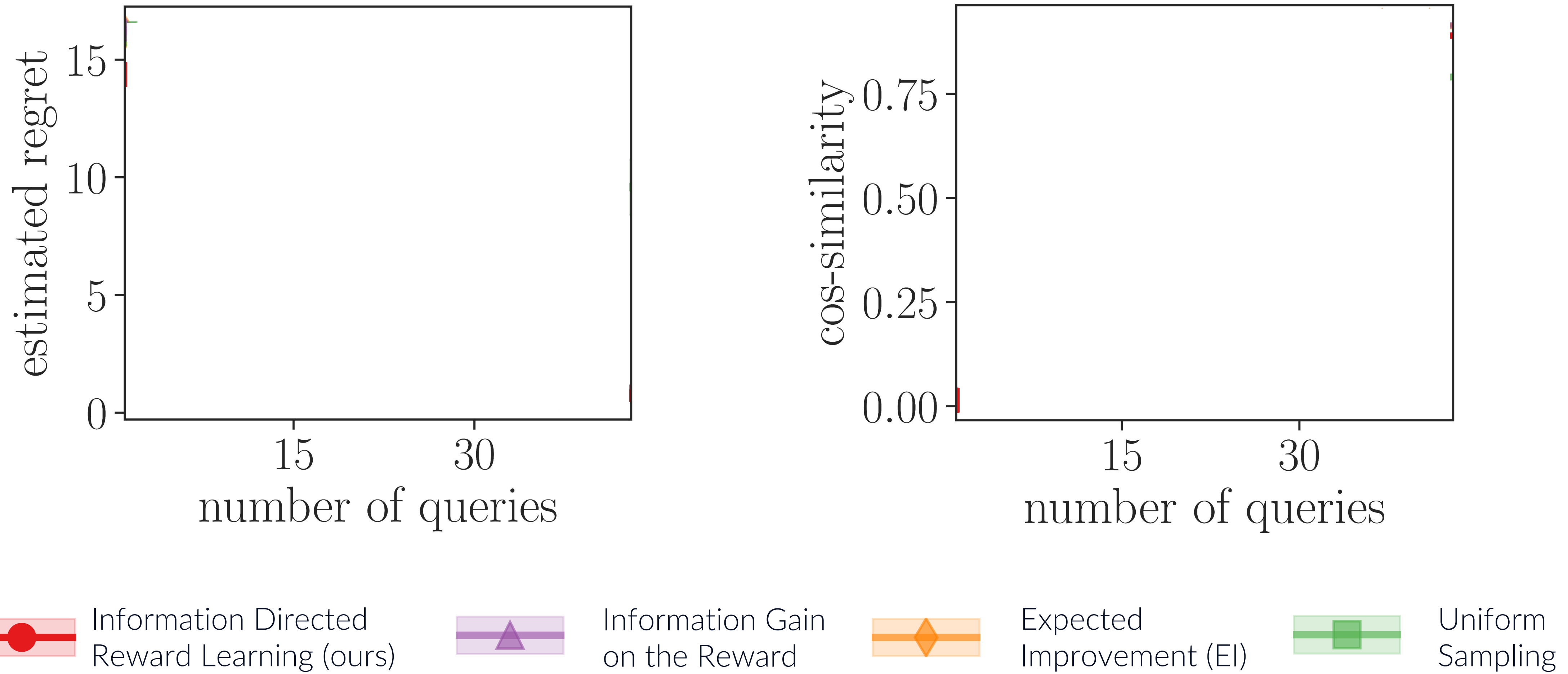
Moving **forward** is rewarded

Goal cell

Moving **backward** is rewarded

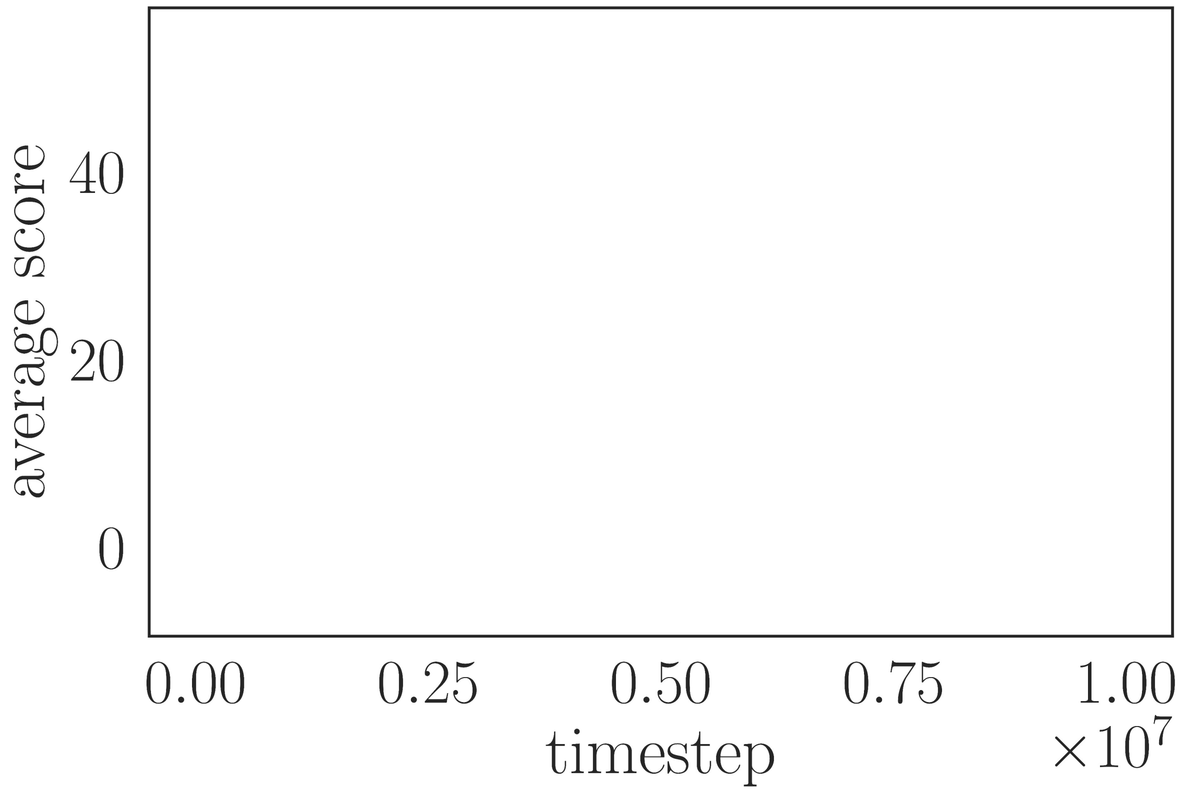
IDRL can learn a complex task in the MuJoCo simulator from numerical evaluations of clips of trajectories

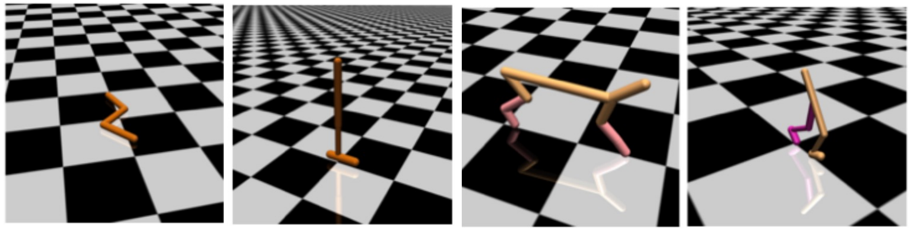
GP reward model



IDRL can learn locomotion in MuJoCo from comparisons

DNN reward model

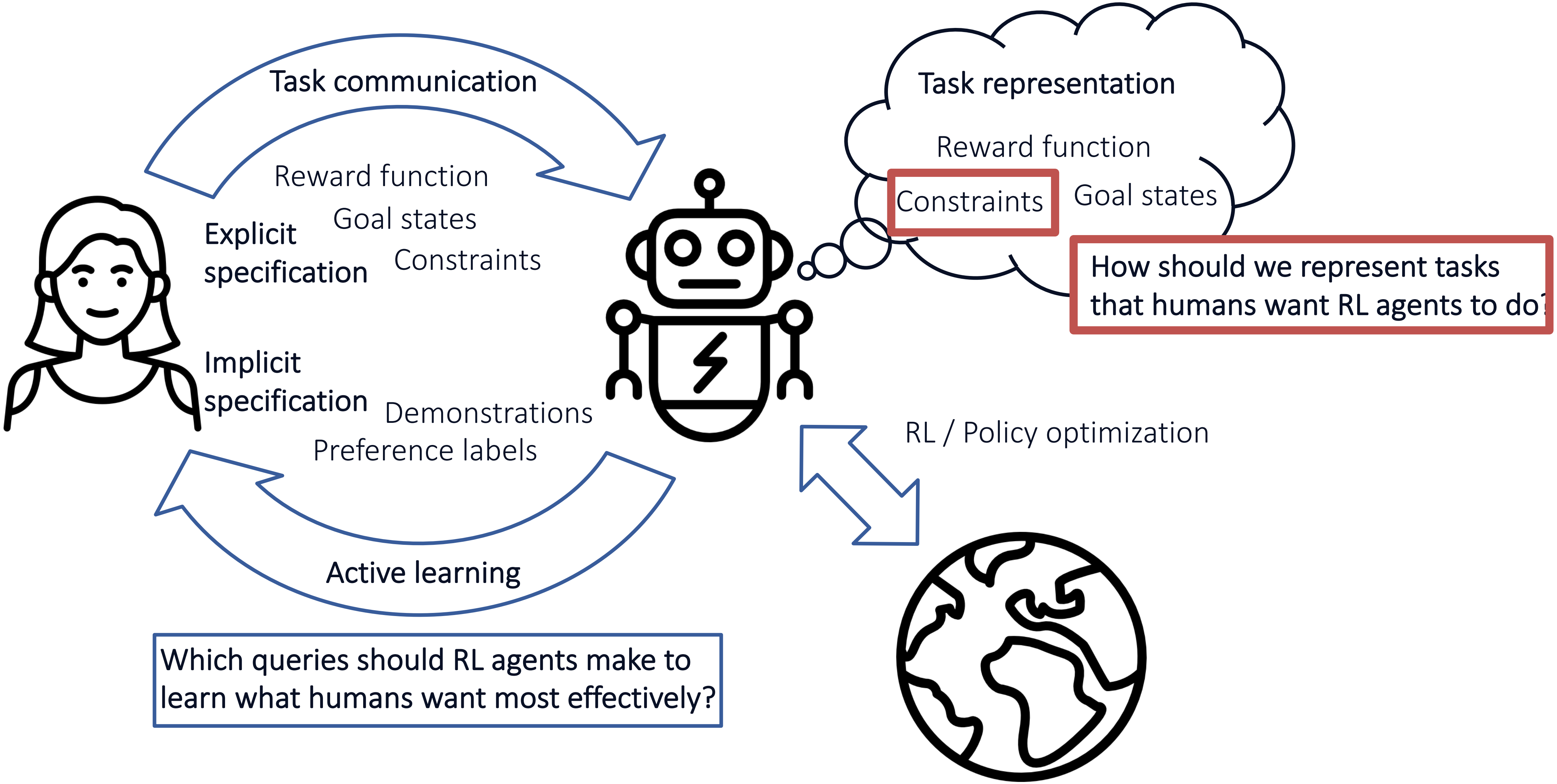




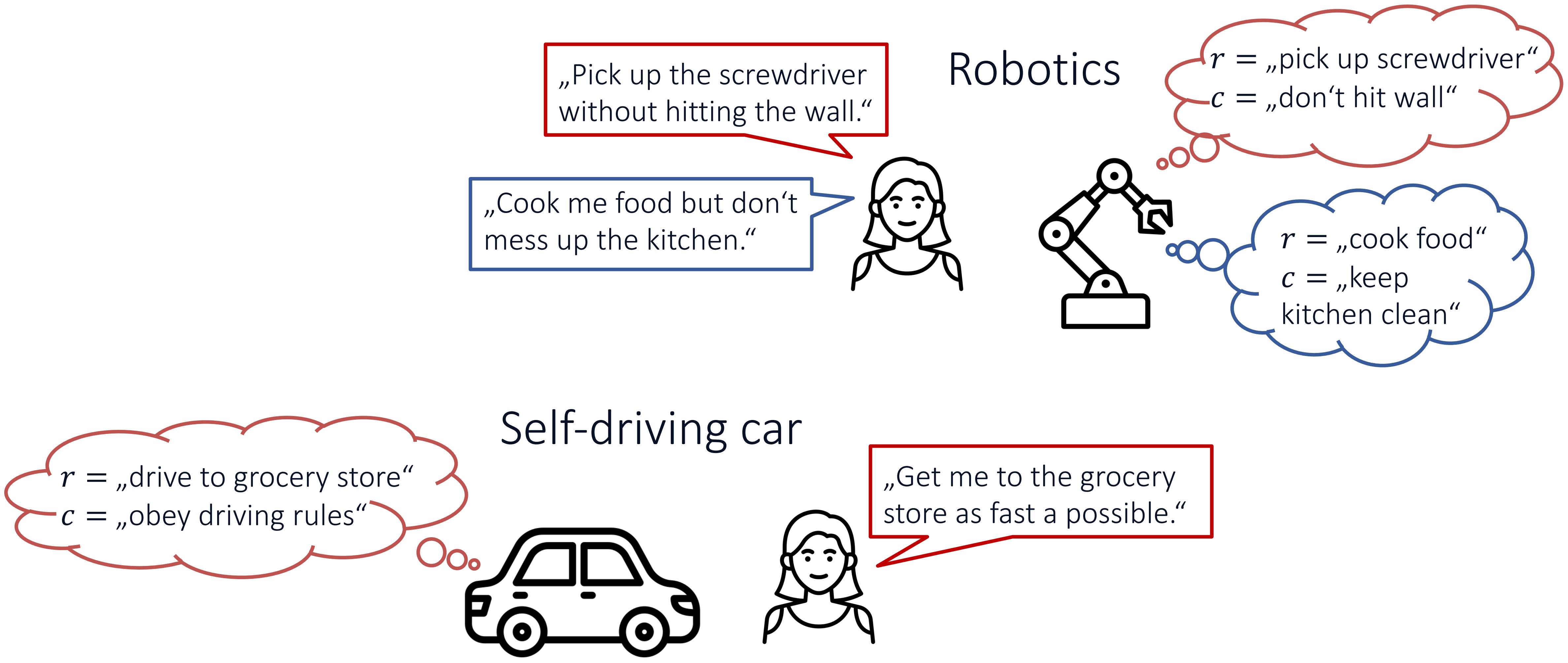
- Normalized score averaged over multiple MuJoCo environments as a function of policy training steps
- Reward model trained from synthetic comparison queries
- Samples provided following a pre-defined schedule that is the same for all methods (more samples initially, less later)

Information Directed Reward Learning (ours) | IDRL w/o candidate policy rollouts | Information Gain on the Reward | Uniform Sampling

How to communicate task specifications to RL agents?



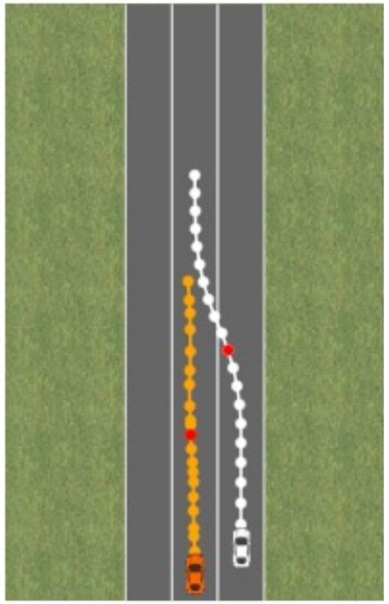
Many practical tasks naturally decompose into rewards and constraints



Constraint have can be more transferable and robust than rewards

Encode task as reward + penalty

Base Scenario



Encode task as reward + constraint

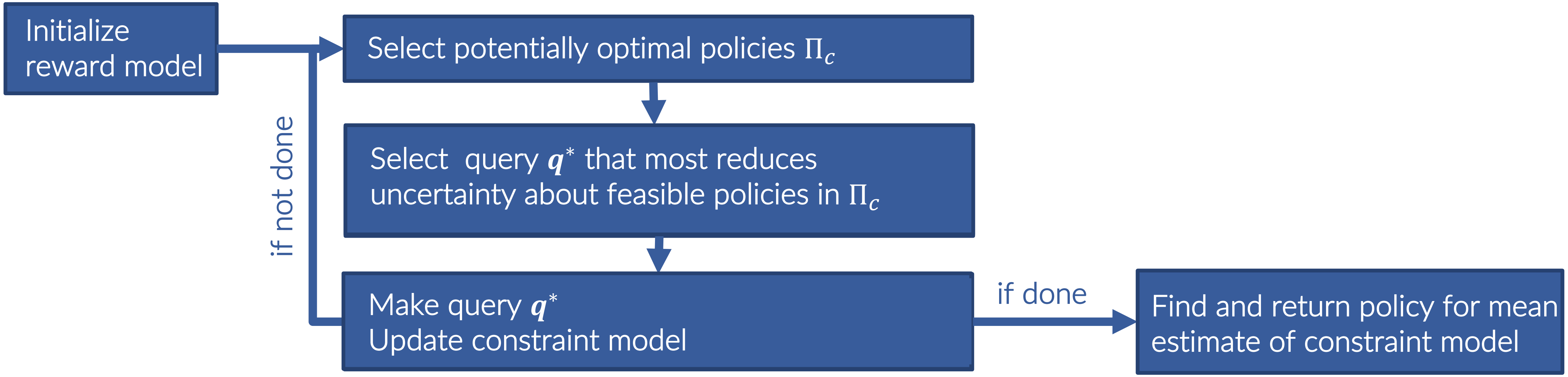
We can actively learn constraints similarly to learning rewards

Information directed reward learning

1. Which policies are potentially optimal?
2. How can we reduce uncertainty about their *optimality*?

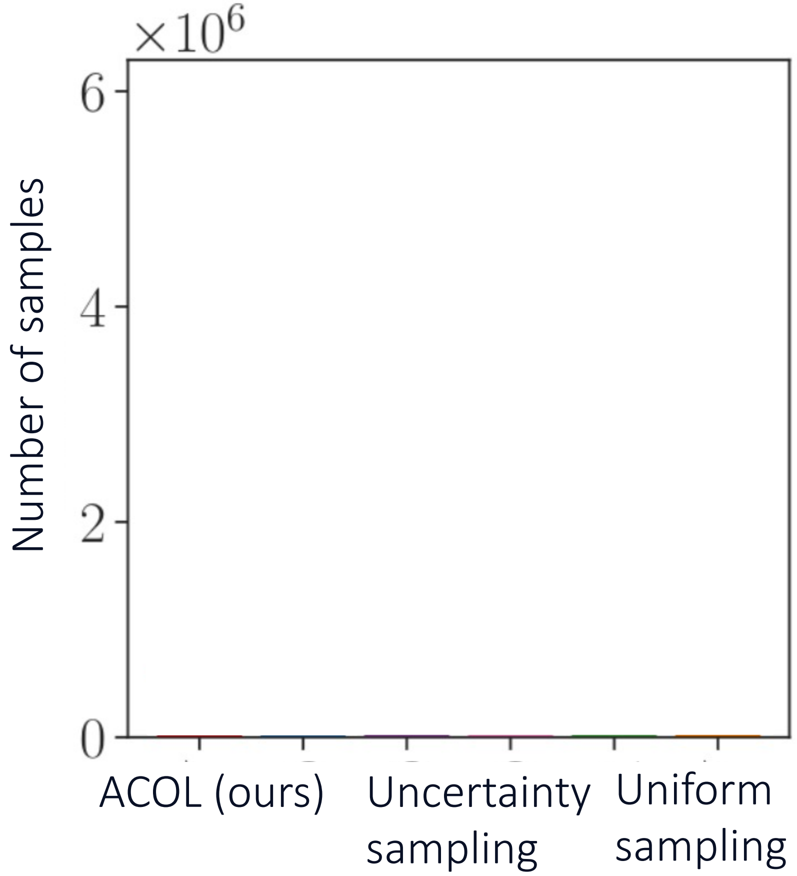
Adaptive constraint learning (ACOL)

1. Which policies are potentially optimal?
2. How can we reduce uncertainty about their *feasibility*?



Active learning also improves sample efficiency for learning constraints

- We consider a driving environment with known reward but unknown constraints
- We can obtain binary samples wheather a trajectory is feasible or not
- How many samples to we need until we can identify the best constrained policy?



Key takeaways

- For active reward learning, we should consider plausibly optimal policies
- Information directed reward learning (IDRL) provides a way to do so

- Rewards and constraints can be a **particularly robust** alternative to represent task specifications compared to reward functions only
- We can actively learn constraints with a similar method as IDRL

https://arxiv.org/abs/2102.12466

Come talk to me if…

- … you want to hear more technical details about anything I talked about.
- … you work on an application that would benefit from learning task specifications from humans.
- … you just want to chat.